Journal of KIIT. Vol. 16, No. 3, pp. 17-24, Mar. 31, 2018. pISSN 1598-8619, eISSN 2093-7571 17
http://dx.doi.org/10.14801/jkiit.2018.16.3.17

비유사성 측정을 위한 준거리 연산

강태원*, 김영희**, 김용찬***

Quasi-distance Operation for Measuring Dissimilarity

Tae-Won Kang*, Young-Hee Kim**, and Yong-Chan Kim***

이 논문은 2016년도 광운대학교 교내학술연구비 지원에 의해 연구되었음

요 약

분류, 군집화, 추천 같은 다양한 데이터 분석은 데이터 항목의 비유사성 즉, 거리를 측정하는 것으로 시작한다. 데이터 집합의 거리관계는 항목들의 거리를 나타내는 거리행렬로 나타낼 수 있는데, 어떤 데이터 집합은 거리행렬을 구하기 어렵다. 준거리 조건은 삼각부등식을 만족하는 것으로 거리척도의 최소조건이다. 이 논문은 주어진 데이터 집합에서 준거리 관계를 생성하는 행렬연산 min_sum을 제안하고, 이 연산과 일반적인 거리척도를 위한 조건과의 관계에 대한 정리를 생성하였다. 그리고 이 연산이 데이터 집합으로부터 직관적으로 만든 잠재적 거리행렬을 준거리 조건을 만족하는 행렬로 변환함을 증명하였다. 의원들의 1년치 실제 투표 데이터를 사용한 투표성향 군집화에 적용한 결과 거리척도로서 효과적임을 확인하였다.

Abstract

Various data analysis, such as classification, clustering, and recommendation, begins by measuring the dissimilarity or distance of data items. The distance relation of the data set can be represented by a distance matrix indicating the distance of the items, but it is difficult to get the distance matrix of some data set. The quasi distance condition satisfies the triangular inequality and is the minimum condition of metric. We propose a matrix operation min_sum that generates a quasi-distance relation in a given data set and generated the theorem about the relation between this operation and the condition for general distance metric. Then we prove that this operation transforms the potential distance matrix intuitively created from the data set into a matrix satisfying the quasi-distance condition. It is confirmed that it is effective as a distance measure as a result of applying voting tendency clustering using 1-year actual voting data of lawmakers.

Keywords
dissimilarity, quasi-distance, distance matrix, metric, clustering

* 강릉원주대학교 컴퓨터공학과 교수
- ORCID: http://orcid.org/0000-0002-8125-6686
** 광운대학교 인제니움학부대학 교수(교신저자)
- ORCID: http://orcid.org/0000-0003-3430-5315
*** 강릉원주대학교 수학과 교수
- ORCID: http://orcid.org/0000-0002-1742-4606

· Received: Feb. 05, 2018, Revised: Mar. 05, 2018, Accepted: Mar. 08, 2018
· Corresponding Author: Young-Hee Kim
Ingenium College of Liberal Arts-Mathematics, Kwangwoon University, Seoul 01897, Korea
Tel.: +82-2-940-5221, Email: yhkim@kw.ac.kr

Ⅰ. 서 론

군집화를 포함한 많은 데이터 분석에 비유사성 측정이 사용된다[1][2]. 흔히 비유사성(Dissimilarity), 유사성, 거리, 근접성(Proximity) 등을 구분하지 않고 사용하지만 각각은 다소 다른 의미를 갖는다. 거리는 떨어진 정도를 나타내므로 유사성을 직접 나타내지 않는다. 유사성 s는 거리 r의 역수를 이용해서 $s=1/(1+r)$ 형태로 사용한다[1][3]. 근접성은 가깝다는 의미를 내포하므로 유사성을 나타내는 다른 말이나 비유사성/유사성을 포괄하는 의미로 종종 사용한다. 비유사성은 대상의 다른 정도를 나타내므로 거리에 의해 직접 측정할 수 있는 것이다.

데이터를 분석하는 행위는 결국 패턴 찾기다. 모든 것이 같다면 아무 할 일이 없다. 데이터 분석의 많은 응용은 각자가 가진 고유한 방법에도 불구하고 거리 즉 비유사성을 측정을 전제로 한다. 덩어리를 찾아내는 군집화, 어디에 속하는지를 찾는 분류는 물론이고 요즘 많이 연구되는 추천시스템 들은 우선 거리를 측정하는 것에서 시작한다. 그런데 거리를 측정하는 것이 항상 가능한 것은 아니다. 예를 들어 두 사람의 취향이 비슷한지를 알려면 거리를 재야하는데, 취향을 잰다는 것이 쉽지는 않다.

본 논문은 비유사성을 측정할 수 있는 거리 연산에 관한 것이다. 먼저 데이터 과학의 제 분야에서 일반적으로 사용하는 거리 연산에 대해 알아보고, 떨어진 정도를 나타내는 거리 척도를 위한 수학적 정의와 조건이 완화된 거리에 대해 소개한다. 다음으로 주어진 데이터에서 준거리 관계를 생성하는 행렬연산 min_sum을 제안하고 증명한다. 이를 데이터 군집화에 적용하여 비유사성 측정 수단으로서 효과를 정성적으로 평가한다. 제안한 준거리 연산을 사용하는 경우 국가 사이의 교역량, 사람들 사이에 주고받은 문자 횟수 등 일반적으로 거리를 정의하기 어려운 데이터에 거리 개념을 부여하여 분류 및 군집화 등을 수행할 수 있다.

Ⅱ. 준거리 생성을 위한 행렬연산 min_sum

이 장에서는 거리에 대한 일반적인 정의와 비유사성 측정을 위해 흔히 사용하는 거리함수에 대해 설명한다. 다음으로 새로운 행렬연산 min-sum을 제안하고 이것이 준거리 조건을 만족하는 거리행렬을 생성함을 증명한다.

2.1 거리에 대한 정의

분류, 군집화, 추천시스템 등 데이터 분석의 많은 분야는 대상의 비유사성을 측정하기 위해 다양한 거리 척도를 사용한다. 많은 응용에서 사용하는 거리 척도는 다음과 같으며 이들의 일반적인 특징에 대해 오랫동안 연구되어 왔다[4]-[6].

표 1. 거리함수
Table 1. Distance functions

Minkowski	$D_{ij}=\left(\sum_{l=1}^{d}\lvert x_{il}-x_{jl}\rvert^{1/n}\right)^{n}$
Euclidean	$D_{ij}=\left(\sum_{l=1}^{d}\lvert x_{il}-x_{jl}\rvert^{1/2}\right)^{2}$
City-block	$D_{ij}=\sum_{l=1}^{d}\lvert x_{il}-x_{jl}\rvert$
Sup distance	$D_{ij}=\max_{1\le l\le d}\lvert x_{il}-x_{jl}\rvert$
Mahalanobis	$D_{ij}=(\boldsymbol{x}_i-\boldsymbol{x}_j)^T\boldsymbol{S}^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j)$
Pearson*	$r_{ij}=\dfrac{\sum_{i=1}^{d}(x_{il}-\overline{x_i})(x_{jl}-\overline{x_j})}{\sqrt{\sum_{i=1}^{d}(x_{il}-\overline{x_i})^2\sum_{i=1}^{d}(x_{jl}-\overline{x_j})^2}}$
Cosine**	$r_{ij}=\cos\alpha=\dfrac{\boldsymbol{x}_i^T\boldsymbol{x}_j}{\lVert\boldsymbol{x}_i\rVert\lVert\boldsymbol{x}_j\rVert}$

항상 그렇듯이 모든 데이터의 비유사성을 잘 측정하는 거리 척도는 없으며 응용에 따라 다르게 평가된다[2][5][7]. 하지만 이들은 거리를 측정하는 함수로서 모두 일정한 조건을 만족한다. 비유사성을 위한 거리 척도로서 거리함수는 일반적으로 다음과 같이 정의한다.

정의1 집합 X와 함수 $d: X\times X\rightarrow[0,\infty)$가 모든 $x, y, z\in X$에 대하여 다음 네 가지 조건을 만족하면 d를 X위의 거리 혹은 거리함수 라고 한다.

(M1) $d(x,x)=0$
(M2) $d(x,z)\le d(x,y)+d(y,z)$
(M3) $d(x,y)=d(y,x)$
(M4) $d(x,y)=0$이면 $x=y$이다.

함수 d가 M1, M2 두 조건만 만족하면 준거리(Quasi-distance)라 부르고, d가 M1, M2, M3 세 조건을 만족하면 왜곡거리(Pseudo-distance)라 부른다. 준거리와 왜곡거리는 거리의 개념을 더 일반화한 것이다. 특히 준거리는 비유사성 측정을 위한 척도의 최소 조건을 만족한다[5][8]. 표 1에서 앞에서부터 5개는 모두 정의1을 만족하는 거리함수다. 반면에 상관계수 r_{ij}(*)와 $\cos\alpha$(**)는 유사성을 나타내는 것으로 거리가 아니다. 따라서 비유사성 척도로 사용할 때는 다음과 같이 변환한다.

$$D_{ij}=-\log((r_{ij}/2)+0.5) \tag{1}$$

거리함수를 이해하기 위해 몇 가지 사례를 설명한다. 어떤 집합 X에 대하여 $X\times X$위에서 $d(x,y)=\begin{cases}1, & x\neq y\\ 0, & x=y\end{cases}$로 정의된 함수는 정의1을 모두 만족하므로 X위의 거리다. 다음으로 정수의 집합 Z에 대하여 $Z\times Z$위에서 $d(x,y)=\begin{cases}x-y, & x\ge y\\ 1, & x<y\end{cases}$로 정의된 함수는 예를 들어 $d(1,3)\neq d(3,1)$이므로 조건 M3를 만족하지 않아서 거리가 아니다. 하지만 다른 조건들은 모두 성립하므로 준거리이다. 마지막으로 평면 R^2위의 두 점 $\mathbf{x}=(x_1,y_1)$, $\mathbf{y}=(x_2,y_2)$에 대하여 $d(\mathbf{x},\mathbf{y})=|x_1-y_1|$으로 정의된 함수는 M1, M2, M3를 만족하므로 왜곡거리이다. 하지만 서로 다른 두 점에 대하여 $d((2,2),(2,3))=0$이므로 M3를 만족하지 않아 거리가 아니다.

2.2 행렬연산 min_sum

이 논문에서 제안하는 거리연산 min_sum은 다음과 같다.

정의2 집합 $X=\{a_1,\cdots,a_n\}$위에서 정의된 함수 $d: X\times X\to[0,\infty)$에 대하여 행렬 $A=(a_{ij})$, $a_{ij}=d(a_i,a_j)$에 대한 연산 min_sum을 다음과 같이 정의한다. 여기서 $\oplus$은 연산 min_sum을 나타내는 연산자이고 $\bigwedge_{k=1}^{n}$는 $k=1,2,\cdots,n$에서 최솟값을 의미한다.

$$A\oplus A=(c_{ij}),\quad c_{ij}=\bigwedge_{k=1}^{n}(a_{ik}+a_{kj}) \tag{2}$$

앞에서 정의한 연산 $\oplus$에 대해 다음 정리를 얻을 수 있다. 앞으로 필요충분조건(if and only if)은 간단히 *iff* 로 나타낸다.

정리1 정의2의 함수 d는 다음을 만족한다.
(1) $d(a_i,a_i)=0$ *iff* $a_{ii}=0$, $i\in\{1,\cdots,n\}$
(2) $d(a_i,a_j)\le d(a_i,a_k)+d(a_k,a_j)$
iff $a_{ij}\le a_{ik}+a_{kj}$, $k\in\{1,\cdots,n\}$ *iff* $a_{ij}\le c_{ij}$.
(3) $d(a_i,a_j)=d(a_j,a_i)$ *iff* $a_{ij}=a_{ji}$, $i,j\in\{1,\cdots,n\}$
(4) $d(a_i,a_j)=0\Rightarrow a_i=a_j$
iff $a_{ij}=0\Rightarrow i=j$.

증명
(1) 정의2에 의하여 성립한다.
(2) $d(a_i,a_j)\le d(a_i,a_k)+d(a_k,a_j)$이라고 가정하면 $a_{ij}\le a_{ik}+a_{kj}$, $k\in\{1,\cdots,n\}$이다. 따라서 $a_{ij}\le\bigwedge_{k=1}^{n}(a_{ik}+a_{kj})=c_{ij}$이다. 역으로 $a_{ij}\le c_{ij}$이면 $a_{ij}\le a_{ik}+a_{kj}$, $k\in\{1,\cdots,n\}$이다. 따라서 $d(a_i,a_j)\le d(a_i,a_k)+d(a_k,a_j)$이 성립한다.
(3) 정의2에 의하여 $d(a_i,a_j)=d(a_j,a_i)$이면 $a_{ij}=a_{ji}$, $i,j\in\{1,\cdots,n\}$이고, 역도 성립한다.
(4) $d(a_i,a_j)=0$일 때 $a_i=a_j$이라고 가정하자. $a_{ij}=0$이면, $d(a_i,a_j)=0$이므로 $a_i=a_j$이고, $i=j$가 성립한다. 역으로 $a_{ij}=0$일 때 $i=j$가 성립한다고 가정하면, $d(a_i,a_j)=0$이면 $a_{ij}=0$이므로 $i=j$이고, 따라서 $a_i=a_j$이다. **QED**

2.3 연산 $\oplus$를 반복 적용하여 준거리 생성

다음 정리는 행렬연산 $\oplus$를 반복 적용하면 준거

리를 생성할 수 있음을 보여준다.

정리2 집합 $X=\{a_1,\cdots,a_n\}$ 위의 거리함수 $d: X\times X \rightarrow [0,\infty)$가 조건 (M1)을 만족하고 행렬 A가 $A=(a_{ij})$, $a_{ij}=d(a_i,a_j)$이면,

$c_{ij}^2 = \bigwedge_{k=1}^{n}(a_{ik}+a_{kj})$, $A\oplus A=(c_{ij}^2)$,

$c_{ij}^4 = \bigwedge_{k=1}^{n}(c_{ik}^2+c_{kj}^2)$,

$(A\oplus A)\oplus(A\oplus A)=(c_{ij}^4)$,

$\cdots$,

$c_{ij}^{2^m} = \bigwedge_{k=1}^{n}(c_{ik}^{2^{m-1}}+c_{kj}^{2^{m-1}})$,

$A^{2^m}=A^{2^{m-1}}\oplus A^{2^{m-1}}=(c_{ij}^{2^m})$일 때,

$\lim c_{ij}^{2^m}=d_{ij}$, $D^2=D\oplus D=D=(d_{ij})$가 존재한다.

증명

정리1에 의해, 다음이 만족한다.

$$0 \le c_{ij}^2 = \bigwedge_{k=1}^{n}(a_{ik}+a_{kj}) \le a_{ii}+a_{ij}=0+a_{ij}=a_{ij}$$

$$0 \le c_{ij}^4 = \bigwedge_{k=1}^{n}(c_{ik}^2+c_{kj}^2), \le c_{ii}^2+c_{ij}^2=0+c_{ij}^2=c_{ij}^2$$

즉, $c_{ij}^{2^m}$은 감소수열이고 아래로 유계이므로 수렴한다. **QED**

따라서 다음이 성립한다.

$\lim c_{ij}^{2^m}=a_{ij}$ 이면

$\forall m\in N,\ c_{ij}^{2^m}=a_{ij},\ A\oplus A=A$

$\lim c_{ij}^{2^m}=c_{ij}^2$ 이면

$\forall m\in N,\ c_{ij}^{2^m}=c_{ij}^2,\ A^2\oplus A^2=A^2$

$\lim c_{ij}^{2^m}=c_{ij}^4$ 이면

$\forall m \ge 2,\ c_{ij}^{2^m}=c_{ij}^4,\ A^4\oplus A^4=A^4$

…

$\lim c_{ij}^{2^m}=d_{ij}$ 이면

$\forall m\in N,\ c_{ij}^{2^m}\neq c_{ij}^{2^{m+1}},\ D\oplus D=D=(d_{ij})$

정의2와 정리2의 2)에 의해 우리는 거리 조건 M1을 만족하는 행렬에 제안한 min_sum 연산 $\oplus$를 반복적용하면 준거리 관계를 만족하는 행렬 D를 생성할 수 있음을 알았다. 제안한 연산이 비유사성 즉 거리 척도로 사용될 수 있음을 보이기 전에 한 가지 사례를 통해 정리2의 의미를 이해해 보자.

지난 1주일 동안 a_1은 a_2, a_3, a_4에게 7, 3, 4번, a_2는 a_1, a_3, a_4에게 3, 4, 2번, a_3는 a_1, a_2, a_4에게 8, 3, 1번, 마지막으로 a_4는 a_1, a_2, a_3에 5, 2, 1번 전화를 했다고 하자. 보다 친밀한 혹은 가까운 사람에게 자주 전화를 한다고 가정하면 네 사람 사이의 거리를 행렬 F로 나타내는 것을 생각할 수 있다.

$$F=\begin{pmatrix} 0 & \frac{1}{7} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{8} & \frac{1}{3} & 0 & 1 \\ \frac{1}{5} & \frac{1}{2} & \frac{1}{4} & 0 \end{pmatrix} \quad F\oplus F=\begin{pmatrix} 0 & \frac{1}{7} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{8} & \frac{15}{56} & 0 & \frac{3}{8} \\ \frac{1}{5} & \frac{12}{35} & \frac{1}{4} & 0 \end{pmatrix} \tag{3}$$

그러나 행렬 F는 예를 들어 $d(a_3,a_2)=1/3 \not\le d(a_3,a_1)+d(a_1,a_2)=1/8+1/7=15/56$이므로 준거리조차 되지 못한다. 따라서 행렬 F는 주어진 데이터의 거리 관계를 나타내는 행렬로 사용할 수 없다. 그러나 F에 min_sum 연산 $\oplus$를 적용한 $F\oplus F$는 준거리다. 실제로 $F\oplus F$는 정의1의 M1, M2, M4를 만족한다. 따라서 행렬 $F\oplus F$은 비유사성을 측정하는 거리행렬로 사용할 수 있다.

III. 성능 평가 및 분석

비유사성/유사성에 대해서는 주로 통계학의 분야에서 70년대까지 잘 정리되었다. 이어진 연구들은 특정 문제와 관련해서, 데이터 유형 즉, 수치데이터, 이진데이터, 범주형데이터, 혼합형데이터 등을 대상으로 비유사성 척도들을 어떻게 활용하는지 등에 관한 것이다[8][9]. 비유사성 척도를 정량적으로 비

교하기는 어렵다. 비유적으로 사람을 비교하는데 키와 몸무게 중 어떤 것이 유용한지는 어디에 사용하느냐에 달린 문제다. 여기서는 $\oplus$ 연산이 상대적인 거리를 효과적으로 측정한다는 것을 정성적인 방법으로 보인다. 앞의 사례를 잘 살펴보면 직관적으로 a_3는 a_4, a_2보다 a_1과 가깝다고 생각되지만 이를 정량적으로 판단하려면 거리를 구할 수 있어야 한다. 하지만 앞에서 본 것과 같이 주어진 데이터 항목의 거리관계를 나타낼 마땅한 방법이 없다. 대신 제안한 $\oplus$ 연산을 사용하여 준거리를 생성할 수 있음을 알았다.

그림 1. $F \oplus F$의 거리 관계 시각화
Fig. 1. Visualization distance relation in $F \oplus F$

그림 1은 이렇게 구한 행렬 $F \oplus F$의 거리 관계를 MDS(Multi-Dimentional Scaling)을 이용해서 나타낸 것이다. MDS는 비유사성 즉, 거리가 정해진 데이터를 시각적으로 나타내는 일종의 그래프 레이아웃이다[3]. 넓은 의미에서는 높은 차원 데이터의 차원을 낮추는 것인데, 여기서는 평면에 나타내는 것이므로 2차원으로 낮춘다. 즉, 거리행렬을 입력으로 받아서 데이터 들 사이의 거리를 가장 잘 반영하는 좌표를 생성해준다. R 프로그래밍에서 제공하는 cmdscale() 함수를 사용하여 거리관계를 시각적으로 나타냈다. 그림은 a_3와 a_1이 상대적으로 가깝다는 것을 보여준다.

이상으로 $\oplus$ 연산으로 준거리를 생성하고 그것을 시각적으로 나타내 보았다. 이제 더 복잡한 데이터 집합의 군집화에 제안한 연산을 적용해 본다. 사용한 데이터는 미국 상원의 기명 투표 데이터다 (http://www.voteview.com). 미국 상원 의원은 공화당, 민주당, 무소속으로 구성된다. 여기서는 투표 데이터를 군집화 하여 투표성향과 소속정당의 관계를 알아보려고 한다. 즉, 의원들의 법안에 대한 투표 데이터를 이용하여 의원들 사이의 거리를 구한 후 소속정당과 함께 시각적으로 나타내었을 때 군집이 형성되는지 알아보려 한다.

원 데이터의 편집과 클러스터링 과정은 참고문헌 [3] 9장의 절차를 참고하였다. 표 2는 의원-투표 데이터 행렬을 나타낸다. 109번 회기를 기준으로 이 행렬의 크기는 101×644이다. 즉, 의원 101명이 644건의 법안에 투표한 결과다. 여기서 1은 찬성, 2는 반대, 3은 기권을 나타낸다. 어떤 두 의원의 벡터열이 일치할수록 두 의원의 투표성향이 같다고 할 수 있다. 따라서 일치하는 정도를 거리로 나타내면 투표성향이 같은 의원은 가까이 위치할 것이다. 이때 각 의원의 위치를 소속정당으로 표시하면 투표성향이 같은 사람들이 어떻게 분포하는지를 알 수 있다.

표 2. 의원-투표 데이터 행렬
Table 2. Senator-Vote data matrix

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
2	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	2	2
5	1	1	1	1	1	1	1	1	1	1	2	2
6	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1
...												
24	3	3	3	3	3	3	3	3	3	3	3	3
25	1	1	1	1	1	1	1	1	1	1	1	1

[3]에서는 의원들 사이의 거리를 의원-투표 행렬의 행 벡터별 유클리드 거리를 계산하여 측정하였다. 그림 2는 R 프로그램 결과를 나타낸다.

```
> rollcall.rawdist[[9]]
          2         3         4         5         6
3  10.488088
4  14.282857 13.416408
5  12.569805 11.224972 11.575837
6   9.643651 11.532563 13.601471 11.357817
7  17.058722 18.138357 17.291616 17.635192 16.552945
8  19.235384 18.330303 19.026298 17.832555 19.261360
9  19.621417 19.104973 19.467922 18.681542 19.949937
10 22.583180 21.771541 22.891046 22.000000 22.248595
11 21.023796 20.297783 20.976177 19.595918 20.615528
12  9.273618  9.899495 13.784049 11.224972  8.660254
```

그림 2. 의원들 사이의 거리(유클리드)
Fig. 2. Distance of senators(Euclid)

예를 들어 2번 의원(첫 번째)과 3번 의원 사이의 거리가 약 10임을 나타낸다. 또한 거리행렬은 대칭이므로 하삼각행렬로 표시되었음을 알 수 있다.

이제 이렇게 계산한 의원들 사이의 거리를 2차원 평면에 펼쳐놓으면 그들의 소속정당별 분포를 알 수 있다. 그러기 위해서 MDS를 적용하여 각 의원들의 좌표를 구한 결과는 표 3과 같다. 예를 들어 SHELBY의원은 민주당 소속이고 상대적 거리를 고려했을 때 위치좌표는 대략 (7.5, 0.95)임을 나타낸다.

표 3. 의원의 좌표(유클리드)
Table 3. Coordinate of senators(Euclid)

	x	y	name	party	congress
2	8.7362450	0.42153751	SESSIONS	200	109
3	7.5136770	0.95859688	SHELBY	200	109
4	6.6998724	0.34395730	MURKOWSKI	200	109
5	6.6437625	1.10798292	STEVENS	200	109
6	8.5641054	0.57522189	KYL	200	109
7	6.0896127	-4.71928428	MCCAIN	200	109
8	-7.0172577	3.44269627	PRYOR	100	109
9	-7.7803120	2.47838047	LINCOLN	100	109
10	-10.0707513	-1.17945860	BOXER	100	109

이 결과를 시각화 하면 그림 3과 같다. 왼쪽에 민주당, 오른쪽에 공화당 의원이 군집을 형성함을 알 수 있다.

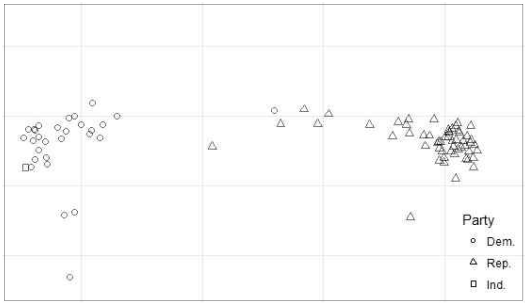

그림 3. 군집(유클리드
Fig. 3. Cluster(Euclid)

제안한 min-sum 연산을 적용하려면 먼저 의원들 사이의(잠재적) 거리를 나타내는 의원-의원 행렬을 구해야 한다. $\boldsymbol{x}-\boldsymbol{y}$ 관계를 나타내는 데이터 행렬 M이 주어지면 M^*M^T와 M^T*M은 각각 $\boldsymbol{x}-\boldsymbol{x}$ 및 $\boldsymbol{y}-\boldsymbol{y}$ 관계를 나타내는 행렬이 된다(여기서 *는 행렬의 곱, M^T는 전치행렬). 따라서 의원-투표 행렬로부터 의원-의원 행렬을 구할 수 있다. 물론 이 행렬은 정의1의 거리조건을 만족하지 않으므로 거리척도로 사용할 수 없다. 기존의 연구에서는 이렇게 얻은 의원-의원 행렬에 대해 행 벡터별 유클리드 거리를 계산하여 거리 행렬을 만드는 번거로운 과정을 거친다[3][7].

이제 의원들 사이의 준거리를 만들어 보자. 의원-투표 행렬을 M이라 하면, 의원-의원 행렬 $N=M^*M^T$을 얻을 수 있다. 행렬 N의 대각성분을 0으로 수정한 행렬을 $N(0)$라 하면 정리2에의해 준거리 $Q=[N(0)\oplus N(0)]^m$를 얻을 수 있다. 실제로 이 경우 $m=1$이다. 그림 4, 표 4, 그림 5는 이렇게 구한 준거리 행렬 Q를 같은 방법으로 MDS로 시각화한 것이다.

```
> view(rollcall.simple[[9]])
```

	V1	V2	V3	V4	V5	V6	V7
1	0	1523	1503	1481	1527	1602	1158
2	1523	0	1505	1487	1497	1573	1165
3	1503	1505	0	1510	1498	1615	1179
4	1481	1487	1510	0	1481	1564	1156
5	1527	1497	1498	1481	0	1596	1143
6	1602	1573	1615	1564	1596	0	1247
7	1158	1165	1179	1156	1143	1247	0

그림 4. 의원들 사이의 거리(min_sum)
Fig. 4. Distance of senators(min_sum)

표 4. 의원의 좌표(min_sum)
Table 4. Coordinate of senators(min_sum)

	x	y	name	party	congress
2	129.20870	10.7349670	SESSIONS	200	109
3	137.68629	15.3810298	SHELBY	200	109
4	136.51909	38.3410705	MURKOWSKI	200	109
5	139.48448	43.8726257	STEVENS	200	109
6	126.23548	18.3591225	KYL	200	109
7	169.58789	-95.2960912	MCCAIN	200	109
8	-185.48704	-7.5457413	PRYOR	100	109
9	-178.20696	-20.2997129	LINCOLN	100	109
10	-154.60453	-50.5728660	BOXER	100	109

그림 3과 마찬가지로 의원들의 투표성향이 소속정당별로 군집을 형성함을 알 수 있다. 즉, 제안한 연산으로 생성한 준거리가 데이터의 특정속성을 분석하기에 적절함을 알 수 있다.

그림 5. 군집(min_sum)
Fig. 5. Cluster(min_sum)

IV. 결 론

분류, 군집화, 추천 같은 데이터 분석의 다양한 분야가 데이터 항목의 비유사성 즉, 거리 측정을 필요로 한다. 데이터 집합의 거리관계는 항목들의 거리를 나타내는 거리행렬로 나타낼 수 있다. 데이터 집합이 항목-값 행렬로 주어지는 경우, 보통 항목벡터들의 유클리드 거리, 상관계수, 코사인 유사도 등을 이용하여 거리행렬을 구하는데, 이는 계산량이 많기도 하고, 직관적이지도 않다. 이 논문에서는 데이터 집합으로부터 직관적으로 만든 잠재적 거리행렬을 준거리 조건을 만족하는 행렬로 변환하는 행렬연산 min_sum을 제안하고, 이 연산을 반복 적용하면 준거리 행렬로 수렴함을 증명하였다. 제안한 연산을 군집화에 적용한 결과 종전의 방법과 같이 거리 척도로서 효과적임을 확인하였다.

이 논문에서는 제안한 행렬연산 min_sum이 거리조건을 만족하지 않는 행렬을 준거리 조건을 만족하는 행렬로 변환함을 증명하고, 그렇게 얻어진 거리행렬이 비유사성을 효과적으로 측정함을 실제 문제에 적용하여 정성적으로 확인하였다. 다음으로 동일한 데이터에 대해 통상의 방법으로 생성한 거리행렬과 제안한 연산을 적용하여 생성한 준거리 행렬을 데이터 유형별로 분석하는 연구와 계산 복잡도를 비교하는 연구가 수행될 수 있겠다.

References

[1] A. D. Gordon, "Classification : Methods for the Exploratory Analysis of Multivariate Data", Measuring Dissimilarity, Chapman and Hall, Chap 2, pp. 13-32, 1981.

[2] Rui Xu and Donald Wunsch II, "Survey of Clustering Algorithms", IEEE Transactions on Neural Networks, Vol. 16, No. 3, pp. 645-678, May 2005.

[3] Drew Conway and John White, "Machine Learning for Hackers", O'Reilly Media, pp. 241, Feb. 2012.

[4] J. C. Gower, "A General Coefficient of Similarity and Some of Its Properties", Biometrics, Vol. 27, No. 4. pp. 857-871, Dec. 1971.

[5] J. C. Gower and P. Legendre, "Metric and Euclidean Properties of Dissimilarity Coefficients", Journal of Classification, Vol. 3, No. 1, pp. 5-48, Mar. 1986.

[6] Guojun Gan, Chaoqun Ma, and Jianhong Wu, "Data Clustering : Theory, Algorithms, and Applications, Similarity and Dissimilarity Measures, ASA-SIAM Series on Statistics and Applied Probability", SIAM, Philadelphia, ASA, Alexandria, VA, pp. 67-106, 2007.

[7] Segaran and Toby, "Programming Collective Intelligence: Building Smart Web 2.0, Applications", O'Reilly Media, pp. 14, 2007.

[8] Brian S. Everitt, Sabine Landau, Morven Leese, and Daniel Stahl, "Cluster Analysis", 5th Ed., John Wiley & Sons, Ltd, pp. 49, 2011.

[9] Charu C. Aggarwal and Chandan K. Reddy, "Data Clustering Algorithms and Applications", CRC Press, pp. 7, 2014.

저자소개

강 태 원 (Tae-Won Kang)

1985년: 연세대학교 수학과 (이학사)
1988년: 고려대학교 전산과학과 (이학사)
1991년: 고려대학교 수학과 (이학석사)
1996년: 고려대학교 컴퓨터학과 (이학박사)
1997년 ~ 현재 : 강릉원주대학교 컴퓨터공학과 교수
관심분야 : 복잡계, 인공생명, 인공지능, 소프트컴퓨팅

김 영 희 (Young-Hee Kim)

1985년 : 연세대학교 수학과 (이학사)
1987년 : 연세대학교 수학과 (이학석사)
1994년 : 연세대학교 수학과 (이학박사)
2003년 ~ 현재 : 광운대학교 인제니움학부대학 교수
관심분야 : 수치해석, 정보수학, 인공지능

김 용 찬 (Yong-Chan Kim)

1982년 : 연세대학교 수학과 (이학사)
1984년 : 연세대학교 수학과 (이학석사)
1991년 : 연세대학교 수학과 (이학박사)
1991년 ~ 현재 : 강릉원주대학교 수학과 교수
관심분야 : 정보수학, 퍼지수학